Hepatoprotective effects of systemic ER activation

BulkRNAseq - Pathway analysis

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(fgsea)
```

```
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#44AA99', '#117733', '#88CCEE', '#332288', '#DDCC77', '#CC6677', '#AA4499', '#882255'),
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
colPals$clusters <- setNames(c('#E6E6E6', '#B3B3B3', '#8C8C8C', '#4D4D4D'),
                             c('1', '2', '3', '4'))
```

Load data

```
# consensus differentially expressed genes
DEGs <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')

# treatment response sets
DEG_sets <- readRDS('results/bulkRNAseq_mmus_DEG_sets.rds')

# RNAseq data
RNAseq <- readRDS('results/bulkRNAseq_mmus_data_filt_norm.rds')

# mmus reactome gene sets from Zhang et al 2020
gene_sets <- list()
gene_sets$reactome <- readGMT('data/reactome_mmus_Zhang2020_annotation.gmt')
```

Gene set enrichment analysis (GSEA)

```
# rank genes by signed -log10 P value
gsea_genes_rnk <- lapply(DEGs$unfilt[c('CDmVsHFDm','DPNVsHFDm','DIPVsHFDm','E2VsHFDm','PPTVsHFDm')], function(x) {
  x %>%
    dplyr::mutate(neg_log10Pval = -log10(pvalue)*sign(log2FoldChange)) %>%
    dplyr::mutate(neg_log10Pval = ifelse(is.na(neg_log10Pval), 0, neg_log10Pval)) %>%
    dplyr::arrange(dplyr::desc(neg_log10Pval))
})

# run GSEA
gsea_res <- lapply(gsea_genes_rnk, function(x) {
  runGSEA(gene.rnk = setNames(x$neg_log10Pval,
                              x$external_gene_name),
          gene.sets = gene_sets$reactome$genesets,
          min.size = 10,
          max.size = 500,
          nproc = 3)
})
```

Build network of perturbed pathways

```
# generate node and edge tables
gsea_net <- buildGSEANetwork(gsea_res, gene_sets$reactome$genesets)

# export node and edge tables for Cytoscape input
# filter for nodes connected by at least 1 edge
nodes <- gsea_net$nodes %>%
  dplyr::filter(id %in% c(gsea_net$edges$source, gsea_net$edges$target))

# write.table(nodes,
#             file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_nodes.tsv',
#             sep = '\t',
#             col.names = TRUE,
#             row.names = FALSE,
#             quote = FALSE)
#
# write.table(gsea_net$edges,
#             file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_edges.tsv',
#             sep = '\t',
#             col.names = TRUE,
#             row.names = FALSE,
#             quote = FALSE)
```

Collapse network to pathway clusters

```
# load cluster assignments from Cytoscape
gsea_net_clusters <- read.table(
  file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_clusters.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE,
  quote = '')

cluster_order <- setNames(seq(1,24),
                          nm = c('Vitamin metabolism', 'Development', 'Hemostasis', 'Phagocytosis',
                                 'Extracellular matrix', 'Carbohydrate metabolism', 'GPCR signaling',
                                 'Lipid metabolism', 'Kinase signaling', 'PI3KAKT regulation',
                                 'Nucleotide metabolism', 'Mitochondrial respiration', 'Transcription',
                                 'Cell cycle', 'Chromatin organization', 'Amino acid metabolism',
                                 'IGF regulation', 'Transport', 'Translation', 'DNA repair',
                                 'Biological oxidations', 'Complement cascade', 'RNA metabolism',
                                 'Adaptive immune system'))

gsea_net_clusters <- gsea_net_clusters %>%
  dplyr::mutate(cl_num = dplyr::recode(cl_label, !!!cluster_order))

# load nodes for reproducibility
nodes <- read.table(
  file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_nodes.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE,
  quote = '')

# generate reduced network
# collapse clusters and correlate according to NES profiles
gsea_net_reduced <- list()

gsea_net_reduced$nodes <- gsea_net_clusters %>%
  dplyr::inner_join(nodes, by = 'id') %>%
  dplyr::group_by(cl_label) %>%
  dplyr::summarise(cluster = cl_num,
                   id = cl_label,
                   n = dplyr::n(),
                   CDmVsHFDm_NES = mean(CDmVsHFDm_NES),
                   DPNVsHFDm_NES = mean(DPNVsHFDm_NES),
                   DIPVsHFDm_NES = mean(DIPVsHFDm_NES),
                   E2VsHFDm_NES = mean(E2VsHFDm_NES),
                   PPTVsHFDm_NES = mean(PPTVsHFDm_NES),
                   genes = paste(genes, collapse = ',')) %>%
  unique() %>%
  dplyr::mutate(genes = paste(unique(unlist(strsplit(genes, ','))), collapse = ',')) %>%
  dplyr::mutate(size = length(unlist(strsplit(genes, ',')))) %>%
  dplyr::ungroup(cl_label) %>%
  dplyr::select(-cl_label) %>%
```

```
  dplyr::arrange(cluster)


comb <- expand.grid(gsea_net_reduced$nodes$id, gsea_net_reduced$nodes$id)
comb <- split(comb, 1:nrow(comb))

gsea_net_reduced$edges <- lapply(comb, function(x) {
  names.NES <- colnames(gsea_net_reduced$nodes)[grep('NES', colnames(gsea_net_reduced$nodes))]
  scores.x <- unlist(gsea_net_reduced$nodes[gsea_net_reduced$nodes$id == x[[1]], names.NES])
  scores.y <- unlist(gsea_net_reduced$nodes[gsea_net_reduced$nodes$id == x[[2]], names.NES])
  cor.res <- cor.test(scores.x, scores.y)

  genes.x <- unlist(strsplit(unlist(gsea_net_reduced$nodes[gsea_net_reduced$nodes$id == x[[1]], 'genes']), ','))
  genes.y <- unlist(strsplit(unlist(gsea_net_reduced$nodes[gsea_net_reduced$nodes$id == x[[2]], 'genes']), ','))

  data.frame(source = x[[1]],
             target = x[[2]],
             interaction = 'correlation',
             correlation = cor.res$estimate,
             pval = cor.res$p.value,
             sign = ifelse(cor.res$estimate>0, 'POS', 'NEG'),
             weight = abs(cor.res$estimate),
             genes = paste(dplyr::intersect(genes.x, genes.y), collapse = ','),
             size = length(dplyr::intersect(genes.x, genes.y)))
}) %>%
  dplyr::bind_rows() %>%
  dplyr::filter(!duplicated(dplyr::select(., -c('source', 'target', 'genes')))) %>%
  dplyr::filter(source != target & weight>0.9) # filter edges to cor >0.9 or <-0.9

# write.table(gsea_net_reduced$nodes,
#             file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_reduced_nodes.tsv',
#             sep = '\t',
#             col.names = TRUE,
#             row.names = FALSE,
#             quote = FALSE)
#
# write.table(gsea_net_reduced$edges,
#             file = 'results/bulkRNAseq_mmus_GSEA_reactome_network_reduced_edges.tsv',
#             sep = '\t',
#             col.names = TRUE,
#             row.names = FALSE,
#             quote = FALSE)
```

Define gene sets for reactome pathway clusters

```
# list of pathway cluster gene sets
pathway_sets <- lapply(setNames(gsea_net_reduced$nodes$id, gsea_net_reduced$nodes$id), function(x) {
  gsea_net_reduced$nodes %>%
    dplyr::filter(id == x) %>%
    dplyr::pull(genes) %>%
    strsplit(',') %>%
    unlist()
})

# check distribution of treatment response sets
DEGs_pathway_distr <- lapply(seq(1:length(pathway_sets)), function(i) {
  data.frame(id = i,
             cluster = names(pathway_sets)[i],
             total_genes = length(pathway_sets[[i]]),
             total_genes_in_background = length(dplyr::intersect(pathway_sets[[i]], RNAseq$annotation$external_gene_name)),
             non_reverted = length(dplyr::intersect(pathway_sets[[i]], DEG_sets$gene_symbols$non_reverted)),
             reverted = length(dplyr::intersect(pathway_sets[[i]], DEG_sets$gene_symbols$reverted)),
             DPN_DIP = length(dplyr::intersect(pathway_sets[[i]], DEG_sets$gene_symbols$DPN_DIP)),
             E2_PPT = length(dplyr::intersect(pathway_sets[[i]], DEG_sets$gene_symbols$E2_PPT)))
}) %>%
  dplyr::bind_rows()

# write.table(DEGs_pathway_distr,
#             file = 'results/bulkRNAseq_mmus_GSEA_reactome_clusters_DEGs_intersection.tsv',
#             sep = '\t',
#             col.names = TRUE,
#             row.names = FALSE,
#             quote = FALSE)
```

Export data

```
# GSEA results and network
# saveRDS(gsea_res, file = 'results/bulkRNAseq_mmus_GSEA_reactome_results.rds')
# saveRDS(gsea_net, file = 'results/bulkRNAseq_mmus_GSEA_reactome_network.rds')

# gene sets for defined reactome pathway clusters
# saveRDS(pathway_sets, file = 'results/bulkRNAseq_mmus_GSEA_reactome_cluster_sets.rds')
```

SessionInfo

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] fgsea_1.14.0    forcats_0.5.1   stringr_1.4.0   dplyr_1.1.2
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.2.1
##  [9] ggplot2_3.3.3   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7          lubridate_1.8.0     lattice_0.20-41
##  [4] snow_0.4-4          assertthat_0.2.1    digest_0.6.27
##  [7] utf8_1.1.4          R6_2.5.1            cellranger_1.1.0
## [10] backports_1.4.1     reprex_2.0.1        evaluate_0.21
## [13] httr_1.4.2          pillar_1.9.0        rlang_1.1.1
## [16] readxl_1.4.0        rstudioapi_0.13     data.table_1.13.6
## [19] Matrix_1.4-1        rmarkdown_2.14      BiocParallel_1.22.0
## [22] munsell_0.5.0       broom_0.8.0         compiler_4.0.5
## [25] modelr_0.1.8        xfun_0.31           pkgconfig_2.0.3
## [28] htmltools_0.5.2     tidyselect_1.2.0    gridExtra_2.3
## [31] fansi_0.4.2         crayon_1.5.1        tzdb_0.3.0
## [34] dbplyr_2.1.1        withr_2.5.0         grid_4.0.5
## [37] jsonlite_1.8.0      gtable_0.3.3        lifecycle_1.0.3
## [40] DBI_1.1.3           magrittr_2.0.3      scales_1.2.1
## [43] cli_3.6.1           stringi_1.5.3       fs_1.5.2
## [46] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.3
## [49] vctrs_0.6.3         fastmatch_1.1-0     RColorBrewer_1.1-3
## [52] tools_4.0.5         glue_1.4.2          hms_1.0.0
## [55] parallel_4.0.5      fastmap_1.1.0       yaml_2.2.1
## [58] colorspace_2.0-0    rvest_1.0.2         knitr_1.31
## [61] haven_2.5.0
```